Cumulative Experience Reflection Paper

Portfolio Assessment (CprE 4940)

Simon Weydert

When I arrived at Iowa State, I was most concerned with my coursework. I'd expected to solely focus on my technical ability as a cybersecurity engineer. However, I came to find that other skills, such as teamwork, ethical strength, and the ability to develop new skills in response to novel problems, were far more important than pure technical know-how.

Learning from Teaching

My most valuable experiences at Iowa State have been as an educator. My time as a teaching assistant for our embedded systems course and a calculus teacher demanded both professional communication and technical competency. Technical communication is challenging; English fluency and technical competency are not givens, so it's not adequate to speak like I would in ordinary conversation. I had to develop a more mindful approach to communication: any piece of technical jargon will need to be clarified; keep colloquialisms sparse.

The most challenging part of those jobs was learning to approach a problem from some else's perspective. Each person understands the same problem a little differently; my responsibility was to identify their position and nudge them in a better direction. The solutions they derived often weren't those I would have liked them to find, and while that was initially frustrating, it served as an important lesson: not everyone thinks like me, and my methods will not work for everyone.

Team Dynamics and Working Smarter

Just as there's a great deal of preparation that goes into being a good teacher, there's just as much work that goes into being a good team member. When I was designing a MIPS processor with a team, keeping an abstracted system-level diagram saved my team an immense amount of time. In my embedded

systems course, my team's final project was ambitious (we emulated a firetruck with a remote-activated spray can and text-to-speech alarm system wired through an external Arduino controlled from a desktop GUI). We avoided an immense amount of heartache and pain by splitting our work into bite-sized tasks that could be worked on independently before jumping into the project.

In a similar fashion, algorithms courses taught me that the start to any great program is not coding, but understanding a problem: creating diagrams to describe a program's ideal behavior and identifying any restrictions and the acceptance criteria the program needs to satisfy (e.g., acceptable runtime) comes before laying a single finger on the keyboard.

Ethical Considerations when Problem Solving

When I do start to work on a project, I need to consider the ethical and moral impacts of its implementation; the applications engineers will undoubtedly have good and bad impacts on the world. For instance, embedded systems can be utilized to rescue people from disaster zones. They can also be used as weapons of war, as drone combat has taken center stage in global conflicts.

I found similar insights in my cryptography course. Blockchain technology has had a negative effect on the environment and the global climate. Instead of ignoring these moral inadequacies, some engineers have endeavored to develop blockchain paradigms that reduce the amount of redundant work that needs to be done for a crypto-system, mitigating the technology's negative effects. There are also positive applications of Blockchain technology, such as smart contracts that make it easier for parties to trust each other. We've seen this help surgeons verify their work in satellite hospitals.

Learning Outside the Classroom

To best strengthen my ability to design sustainable solutions, I need a wide palette of experience within and outside of Iowa State. My most recent venture outside the university was the Vermeer Vehicle Technology Bootcamp. The instructors gave a crash course on the CAN bus: a communication protocol for vehicle computer systems. The remainder of the bootcamp was an opportunity to work alongside a

diverse set of academic minds, since we needed both mechanical and computer engineers to remotely operate a vehicle.

The cyber defense competitions on campus help me grow as an engineer because they expose what I don't know. There are always challenges that I fail to complete, and our systems are always busted in new and interesting ways. It reminds me that I have to keep my mind sharp through a process of continual learning, seeking challenges to my current understanding of the world and my technical knowledge.

Conclusion

The ultimate lesson I've learned at this college is that you never know what you'll need to learn. Many fellow cybersecurity engineers overlooked limits and summations as something that aren't related to their major, and yet they are incredible for assessing the runtime of algorithms. Statistics seem boring when you first learn the subject, but they're an invaluable tool for investigating the strength of a cryptographic protocol or utility. Learning more now, even if it doesn't seem related to your major or profession, is almost always a good thing; it keeps an engineer competent and versatile.